## 20장. XML 레이아웃 리소스로 GridLayout 사용하기

## GridLayout에서 사용할 수 있는 속성들

* android:rowCount : 그리드를 구성하는 행(row)의 개수
* android:columnCount : 그리드를 구성하는 열(column)의 개수
* android:orientation : 방향 정의 (예: horizontal, vertical)

## 자식 뷰에서 사용 할 수 있는 속성들

* android:layout\_row : 자식 뷰가 위치할 부모 그리드의 행 번호
* android:layout\_column : 자식 뷰가 위치할 부모 그리드의 열 번호
* android:layout\_gravity : 부모 컨테이너의 여유 공간에 뷰가 모두 채워지지 않아 여유 공간 안에서 뷰를 정렬할 때(가로|세로 방향 값)
* android:layout\_rowSpan : 행 병합할 개수
* android:layout\_columnSpan : 열 병합할 개수
* android:layout\_margin : 자식 뷰의 둘레의 마진을 설정

## 21장. 안드로이드 이벤트 처리 개요

* 이벤트 큐(event queye)에 의해, 선입선출(FIFO, first-in first-out)
* 터치가 된 화면 위치에 있는 뷰에 이벤트의 통지와 정보 전달됨.
* 뷰에서는 이벤트 리스너(event listener)를 가지고 있어야 함.
* 이벤트 리스너 등록 후, 리스너에 대한 콜백(callback) 메서드를 뷰에서 구현해야 함.

## 22장. 안드로이드 스튜디오의 InstantRun 사용하기

* 변경 결과물을 확인하기 위해 프로젝트 전체를 다시 컴파일 및 APK 패키징, 설치 등을 하지 않아도 됨.
* 리소스 설정, 메서드 코드 변경 시, InstantRun이 해당 부분만을 동적으로 교체(swap)함.
* API 21 이상에서만 변경 부분만을 반영 후 앱 전체가 다시 시작
* API 20 이하 버전의 안드로이드에서는 앱 전체가 다시 빌드 및 설치되고 시작됨.
* 어떤 부분을 변경했는지에 따라 Hot, Warm, Cold 세 가지로 레벨이 결정됨.

## 23장. 터치와 다중 터치 이벤트 처리하기

* OnTouchListener 이벤트 리스너를 등록하고, onTouch() 콜백 메서드를 구현해서 사용
* onTouch() 콜백 메서드는 boolean 값을 반환해야 함.
* boolean 값은 발생된 이벤트를 동일한 뷰에 등록된 그다음 일치하는 이벤트 리스너에게 전달할 필요가 있는지 여부임.
* MotionEvent 객체를 통해 뷰 안의 터치 위치, 수행된 액션 타입을 확인할 수 있음.

## 24장. 안드로이드 제스처 감지 클래스로 일반 제스처 처리하기

* 제스처(geture)란 터치 스키린과 사용자 간의 연속적인 상호작용을 의미함.
* 화면이 최초 터치하는 지점에서 시작됨.
* 손가락이나 포인팅 장치가 화면에서 떨어졌을 때 끝남.
* 사용자와 애플리케이션 간의 의사소통 형태로 구현될 수 있음.
* 일반 제스처와 커스텀 제스처 모두 애플리케이션에서 감지하고 처리 할 수 있음.

## 25장. 커스텀 제스처와 핀치 인식 구현하기

* 커스텀 제스처는 안드로이드 장치의 화면에서 수행되는 어떤 연속된 터치 동작이라도 거의 다 포함될 수 있음
* 제스처 빌더(Gesture Builder) 애플리케이션을 사용해서 생성 가능. (SDK 샘플 패키지 내 존재함.)
* 실제 장치 혹은 에뮬레이터에서 지스처 빌더 애플리케이션 실행 후, 제스처를 그리면 됨.
* 제스처 데이터를 GestureLibrary 클래스의 인스턴스로 로드하여 사용하게 됨.
* 핀치 제스처는 두 손가락을 오므리거나 벌리는 동작을 말함

## 26장. 안드로이드 프래그먼트 개요

* 프래그먼트(Fragment)는 액티비티 내부에서 독립적으로 애플리케이션의 사용자 인터페이스를 관리하는 객체
* 프래그먼트를 조립하여 액티비티를 생성할 수 있음
* 런타임 시에는 사용자 인터페이스를 동적으로 변경하기 위해 액티비티에 추가하거나 제거할 수 있음.

## 27장. 안드로이드 스튜디오에서 프래그먼트 사용하기 - 예제 프로젝트

* 프래그먼트를 사용하는 방법을 확인할 수 있었음
* 프래그먼트 클래스와 레이아웃을 생성하고 프래그먼트를 액티비티에 포함시켰음
* 호스팅 액티비티를 통해 프래그먼트끼리 통신하는 방법을 알 수 있었음.

## 28장. 오버플로 메뉴 생성과 관리

* 오버플로 메뉴는 옵션 메뉴라고도 함
* 사용자가 장치 화면에서 액세스 가능한 메뉴
* 개발자가 애플리케이션의 다른 옵션(선택사항)을 포함시킬 수 있음
* 안드로이드 버전에 따라 메뉴의 위치가 달라짐

## 29장. 안드로이드 전환 프레임워크

* 안드로이드 4.4 킷캣(KitKat)에서 소개됨
* 화면을 구성하는 뷰에 애니메이션을 쉽게 추가할 수 있음
* 장면(Scene) : 어느 한 시점의 프레임
* 전환(Transition) : 장면이 바뀌는 것
* 인터폴레이터(Interpolator) : 사전 정의된 다양한 방법으로 애니메이션을 변경할 수 있게 해줌

## 30장. beginDelayedTransition을 사용한 안드로이드 전환

* 한 장면(scene)에서 다른 장면으로 전환하기 위해 레이아웃 파일과 전환 리소스 파일 사용.
* 두 장면 간의 전환 애미네이션 효과를 구성하기 위한 전환 XML 리소스 파일을 사용해 봄.